Concurrent Quantum Computation

F. Yamaguchi*, C. P. Master* and Y. Yamamoto*,†

*Quantum Entanglement Project, ICORP, JST

Edward L. Ginzton Laboratory, Stanford University, Stanford, CA 94305, USA

†NTT Basic Research Laboratories

3-1 Morinosato-Wakamiya, Atsugi, Kanagawa 243-0198, JAPAN

Abstract

A quantum computer is a multi-particle interferometer that comprises beam splitters at both ends and arms, where the n two-level particles undergo the interactions among them. The arms are designed so that relevant functions required to produce a computational result is stored in the phase shifts of the 2^n arms. They can be detected by interferometry that allows us to utilize quantum parallelism. Quantum algorithms are accountable for what interferometers to be constructed to compute particular problems. A standard formalism for constructing the arms has been developed by the extension of classical reversible gate arrays. By its nature of sequential applications of logic operations, the required number of gates increases exponentially as the problem size grows. This may cause a crucial obstacle to perform a quantum computation within a limited decoherence time. We propose a direct and concurrent construction of the interferometer arms by one-time evolution of a physical system with arbitrary multi-particle interactions. It is inherently quantum mechanical and has no classical analogue. Encoding the functions used in Shor's algorithm for prime factoring, Grover's algorithm and Deutsch-Jozsa algorithm requires only one-time evolution of such a system regardless of the problem size n as opposed to its standard sequential counterpart that takes $O(n^3)$, $O(n)$ and $O(n2^n)$.

A computation entails encoding of a function whether classically or quantum mechanically. The encoding of a function has been carried out in the form of a bit-flip. Such an "oracle" $U_{\rm c}$ in reversible classical computers and also in proposed quantum computers is a transformation of an n-bit input x and a work bit w,

$$U_{\rm c} : |x\rangle\,|w\rangle \rightarrow |x\rangle\,|w \oplus f(x)\rangle\,, \tag{1}$$

where $\oplus$ stands for exclusive-OR. When the work bit is initially set to be 0, the oracle returns the function value $f(x)$ in the work bit. The customary construction of such an oracle pertains to sequential application of reversible gates[1]. In quantum computation, however, the oracle is often converted into a transformation of the form,

$$U_{\rm q} : |x\rangle \rightarrow (-1)^{f(x)}\,|x\rangle\,, \tag{2}$$

where the information about $f(x)$ is encoded in the phase of a linear superposition state so that quantum parallelism could be utilized[2,4,6,7]. The conversion of $U_{\rm c}$ into $U_{\rm q}$ is performed by supplementary transformation[2] or automatically by appropriately initialized work bit[3]. The construction of the oracle (1) by sequential application of one-bit and two-bit gates demand an exponential or polynomial number of operations as the problem size grows as shown in Table 1. In order to perform a quantum computation, we need to complete the construction of the oracle while the coherence of the system is maintained, although classical computation does not bring the issue of a limited decoherence time into a question. We propose a concurrent construction of the transformation $U_{\rm q}$ by only one-time evolution of a physical system that has arbitrary multi-particle interactions. The exponentially hard work that is expressed by the complexity of gate arrays to be applied to a physical system in the case of the sequential construction will be replaced by adjustment of an exponentially large number of coupling strengths in the system before the coherence of the system is required.

The system required to concurrently implement an arbitrary n-bit Boolean function $f(x)$ for an n-bit string x consists of n two-level particles with arbitrary multi-particle interactions among them. The Hamiltonian of the system is,

$$\mathcal{H} = \sum_{i=1}^{n} \hbar\omega_i \sigma_{iz} + \sum_{i<j} \hbar\omega_{ij} \sigma_{iz} \otimes \sigma_{jz} + \sum_{i<j<k} \hbar\omega_{ijk} \sigma_{iz} \otimes \sigma_{jz} \otimes \sigma_{kz} + \cdots + \hbar\omega_{12\cdots n} \sigma_{1z} \otimes \sigma_{2z} \otimes \cdots \otimes \sigma_{nz}, \tag{3}$$

where $\otimes$ stands for tensor product. The eigenstates of the Pauli matrix $\sigma_{iz} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ for the i-th particle is used as the computational basis $\{|0\rangle, |1\rangle\}$ for the i-th bit x_i, where $\sigma_{iz} |x_i\rangle = (-1)^{x_i} |x_i\rangle$ ($x_i = \{0, 1\}$). The computational basis of x is defined in the 2^n-dimensional state space spanned by the two-basis states of the n particles, $|x\rangle = |x_n \cdots x_1\rangle = |x_n\rangle \oplus \cdots \oplus |x_1\rangle$. The state $|x\rangle$ represents a number $x = \sum_{i=1}^{n} 2^{i-1} x_i$. The Hamiltonian is diagonal in the computational basis $\{|00\cdots0\rangle, |00\cdots1\rangle, \cdots, |11\cdots1\rangle\}$. Diagonal elements of the $(2^n - 1)$ terms in the Hamiltonian and a 2^n-dimensional vector $(1, 1, \cdots, 1)$ (diagonal elements of the $2^n \times 2^n$ identity matrix) constitute an orthonormal basis to expand 2^n-dimensional vectors. It suggests that the transformation $U_{\rm q}$ in (2), which is a $(2^n \times 2^n)$-diagonal matrix, can be constructed by a global phase shift and one-time evolution of the system, $U_{\mathcal{H}}(\tau) = e^{-\frac{i}{\hbar}\mathcal{H}\tau}$ for time τ as

$$U_{\rm q} = e^{i\phi} U_{\mathcal{H}}(\tau). \tag{4}$$

The concurrent construction of the transformation $U_{\rm q}$ for a given function comprises a preparation of the Hamiltonian (by adjusting the coefficients, $\hbar\omega_i$, $\hbar\omega_{ij}$, $\cdots$, $\hbar\omega_{12\cdots n}$) so that (4) is satisfied and a time-evolution by that Hamiltonian. Only the latter process requires the conserved quantum coherence of the system.

The diagonal elements of (4) are decomposed as

$$e^{i\pi f(x)} = e^{i\phi} \times \prod_i e^{-i\omega_i \tau (-1)^{x_i}} \times \prod_{i<j} e^{-i\omega_{ij}\tau(-1)^{x_i + x_j}} \times \prod_{i<j<k} e^{-i\omega_{ijk}\tau(-1)^{x_i+x_j+x_k}} \times \cdots \times e^{-i\omega_{12\cdots n}\tau(-1)^{x_1+x_2+\cdots x_n}}, \tag{5}$$

which gives the solution,

$$\begin{aligned}
\phi &= \frac{\pi}{2^n}\sum_x f(x), \\
\omega_i\tau &= -\frac{\pi}{2^n}\sum_x (-1)^{x_i} f(x), \\
\omega_{ij}\tau &= -\frac{\pi}{2^n}\sum_x (-1)^{x_i+x_j} f(x), \\
&\vdots \\
\omega_{12\cdots n}\tau &= -\frac{\pi}{2^n}\sum_x (-1)^{x_1+x_2+\cdots+x_n} f(x),
\end{aligned} \tag{6}$$

and determines the coefficients in the Hamiltonian to be prepared so that only one-time evolution for time τ by the Hamiltonian will construct $U_{\rm q}$ by itself for a given function $f(x)$. Note that when $f(x)$ is a constant function, all coefficients in (6) except ϕ are zero.

The formalism for concurrently constructing the transformation $U_{\rm q}$ can be applied to encode functions and calculated values that are used in existing quantum algorithms. Examples are shown below.

Deutsch-Jozsa algorithm[2,4,5]. The algorithm solves the following problem by quantum parallelism. Given the oracle for an n-bit Boolean function $f : \{0,1\}^n \to \{0,1\}$, determine either (A) f is a constant function (at 0 or 1) or (B) f is a balanced function (the sequence $f(0)$, $\cdots$, $f(2^n-1)$ contains exactly 2^{n-1} zeros and 2^{n-1} ones). In the original algorithm, the oracle that has a form of $U_{\rm c}$ is used twice together with another unitary operation on the work bit $S : |w\rangle \to (-1)^w |w\rangle$ between the two applications. The three unitary operations applied to the system in series are designed so that the information about $f(x)$ is transferred from the work bit to the phase of the control register $|x\rangle$ as in (2). Now we have a method to construct the transformation $U_{\rm q}$ concurrently, and the work bit can be removed.

When $f(x)$ takes only 0 or 1, $f(x)$ in (6) can be replaced by $-f(x)$ since $\pi f(x) \equiv -\pi f(x) \pmod{2\pi}$. Therefore a set of parameters obtained by replacing $f(x)$ by $(-1)^{x_1+x_2+\cdots+x_n}(f(x)-2N_x)$ (N_x is an integer) in (6) is also solution to (5). For a balanced function, we choose $N_x = x_1x_2$, so that $\omega_{12\cdots n}\tau = -\frac{\pi}{2^n}\left(2^{n-1} - \sum_x 2N_x\right) = 0$. Thus, to encode a balanced n-bit Boolean function, we need the multi-particle interactions up to

$(n-1)$-th order (the n-particle interaction is not required). Other parameters are determined as,

$$\begin{aligned}
&\phi = \frac{\pi}{2^n}\sum_x (-1)^{x_1+x_2+\cdots+x_n}(f(x)-2N_x),\\
&\omega_i\tau = -\frac{\pi}{2^n}\sum_x (-1)^{x_1+\cdots+x_{i-1}+x_{i+1}\cdots+x_n}(f(x)-2N_x),\\
&\qquad\vdots
\end{aligned} \tag{7}$$

Grover's algorithm[6,7]. The algorithm explains how a data can be found out of 2^n random data entries. The data search problem is described by a function $f(x)$ that returns 1 for a single unknown value of x, say $x=\tau$, and 0 for the rest of x. The algorithm uses the information about $f(x)$ encoded in phase of the control register as in (2), which can be implemented concurrently, when the coefficients in the Hamiltonian are chosen as $\phi = \frac{\pi}{2^n}$, $\omega_i\tau = -\frac{\pi}{2^n}(-1)^{\tau_i}$, $\omega_{ij}\tau = -\frac{\pi}{2^n}(-1)^{\tau_i+\tau_j}$, $\cdots$, and $\omega_{12\cdots n}\tau = -\frac{\pi}{2^n}(-1)^{\tau_1+\tau_2+\cdots\tau_n}$.

(iii) Simon's algorithm[8] determines whether a given function $f:\{0,1\}^n \to \{0,1\}^m$ with $m \geq n$ is periodic $f(x)=f(x') \leftrightarrow x' = x\oplus s$ (with a nontrivial string s) or one-to-one. Encoding the function $f(x)$ in the oracle of the form $U_{\rm c}$ in the original algorithm can be replaced by

$$|x\rangle \to e^{i\frac{\pi}{2^{m-1}}f(x)}|x\rangle\,, \tag{8}$$

which can be constructed concurrently. The transformation (8) has the same form as $U_{\rm q}$, where $(-1)=e^{i\pi}$ is replaced by $e^{i\pi/2^{m-1}}$. The formalism for constructing the transformation $U_{\rm q}$ works exactly the same way. In order to implement an arbitrary function in this problem, all multi-particle interactions up to n-particle interaction in the Hamiltonian are necessary.

Shor's algorithm for prime factorization[9]. In order to factorize an odd number N, we randomly choose a (N and a need to be relatively prime) and find the order r of a, the least r that satisfies $a^r \equiv 1 \bmod N$. Finding the order is the prime part of Shor's algorithm and starts with the transformation on two n-bit registers,

$$\frac{1}{\sqrt{q}}\sum_{x=0}^{q-1}|x\rangle|1\rangle \to \frac{1}{\sqrt{q}}\sum_{x=0}^{q-1}|x\rangle|a^x \bmod N\rangle\,, \tag{9}$$

where $q = 2^n$ satisfies $N^2 \leq q < 2N^2$. Replacing the transformation (9) by,

$$\frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle \rightarrow \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} e^{i\frac{\pi}{2N}(a^x \bmod N)} |x\rangle \,, \tag{10}$$

which encodes $a^x \bmod N$ in the phase of the control register instead of in the work bit, leaves the algorithm unchanged[10]. The phase factors $a^x \bmod N$ in (10) can be calculated classically as,

$$e^{i\frac{\pi}{2N}(a^x \bmod N)} = e^{i\frac{\pi}{2N}\prod_{i=1}^{n}(a^{2^{i-1}} \bmod N)^{x_i}}, \tag{11}$$

where products refer to multiplication mod (N). Equation (11) has the same form as (5) if $\lambda_i^{x_i} = \frac{1+\lambda_i}{2} + \frac{1-\lambda_i}{2}(-1)^{x_i}$ $(\lambda_i = a^{2^{i-1}} \bmod N)$ is used. Therefore, the time-evolution of the system for time τ constructs the transformation (11) by itself when the coefficients are chosen as, $\phi = \prod_{i=1}^{n} \frac{1+\lambda_i}{2}$, $-\omega_i\tau = \phi \times \frac{1-\lambda_i}{1+\lambda_i}$, $-\omega_{ij}\tau = \phi \times \frac{1-\lambda_i}{1+\lambda_i} \times \frac{1-\lambda_j}{1+\lambda_j}$, $\cdots$, and $-\omega_{12\cdots n}\tau = \prod_{i=1}^{n} \frac{1-\lambda_i}{2}$.

Quantum Fourier transform. To perform Quantum Fourier transform A_q for an n-bit register $(q = 2^n)$,

$$A_q : |x\rangle \rightarrow \sum_{y=0}^{q-1} e^{2\pi i x y/q} |y\rangle \,, \tag{12}$$

as used in Shor's algorithm, we need the Walsh-Hadamard transformation on each bit, $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ in the computational basis, and controlled-phase-shift operators on pairs of bits, defined as $S_{j,k} = e^{i\theta_{k-j} x_j x_k}$ on the j-th bit and k-th bit with $j < k$ where $\theta_{k-j} = \pi/2^{k-j}$. The right-hand side of (5) is equal to $S_{j,k}$ if we choose $2\omega_{jk}\tau = \theta_{k-j}$, $2\omega_j\tau = -\theta_{k-j}$, $2\omega_k\tau = -\theta_{k-j}$ and the rest of the parameters are zero, aside from the global phase factor. A product of multiple controlled-phase-shift operators $S_{l,l'} S_{l,l'-1} \cdots S_{l,l+1}$ is also diagonal in the computational basis and can be implemented concurrently by setting $2\omega_{l,m}\tau = \theta_{m-l}$ $(m = l+1, \cdots, l')$, $2\omega_m\tau = -\theta_{m-l}$, $2\omega_l\tau = -\sum_m \theta_{m-l}$. One-bit rotations and two-particle interactions in the Hamiltonian are sufficient to implement controlled-phase-shift operators and products of those concurrently.

The time evolution of a system that concurrently implements functions by our proposed scheme and the number of necessary elementary gates for the function implementation by

means of the standard sequential scheme are compared in Table 1 for existing quantum algorithm. It is challenging to find a system that has multi-particle interactions with reasonably large and controllable strengths in order to implement arbitrary functions, but if found, many functions and quantum algorithms will be implemented by only one-time evolution of the system, which may cross out the current biggest obstacle to quantum computation, the short decoherence time of a quantum system.

Algorithm/Function	Concurrent implementation		Sequential implementation
	Terms in the Hamiltonian	Evolution time	Number of elementary gates
	$U_{\mathrm{q}}: \|x\rangle \to (-1)^{f(x)} \|x\rangle$		
General Boolean function $f(x)$	$RI^{(2)} \cdots I^{(n)}$	$O(1)$	$O(n2^n)$
Deutsch-Jozsa	$RI^{(2)} \cdots I^{(n-1)}$		
Grover	$RI^{(2)} \cdots I^{(n)}$		$O(n)$
	$\|x\rangle \to \exp\left[i\varphi f(x)\right] \|x\rangle$		$\|x\rangle \|0\rangle \to \|x\rangle \|f(x)\rangle$
Shor ($\varphi = \frac{\pi}{2N}$, $f(x) = a^x \bmod N$)	$RI^{(2)} \cdots I^{(n)}$	$O(1)$	$O(n^3)$
Simon ($\varphi = \frac{\pi}{2^{m-1}}$)			$O(mn2^n)$
	$S_{j,k}: \|x\rangle \to \exp\left[i\frac{\pi}{2^{k-j}} x_k x_j\right] \|x\rangle$		
controlled-phase-shift $S_{j,k}$	$RI^{(2)}$	$O(1)$	$O(1)$
	$H_n S_{n-1,n} H_{n-1} S_{n-2,n} S_{n-2,n-1} H_{n-2} \cdots H_2 S_{1,n} S_{1,n-1} \cdots S_{1,3} S_{1,2} H_1$		
Quantum Fourier Transform	$RI^{(2)}$	$O(n)$	$O(n^2)$

Table 1.

Scaling laws of evolution time and number of elementary gates necessary to implement functions for existing algorithms by means of the proposed concurrent implementation and the standard sequential implementation, respectively. In the most general case, an n-bit Boolean function $f(x) : \{0,1\}^n \to \{0,1\}$ is encoded in phases of the control register $|x\rangle$, as $U_{\rm q} : |x\rangle \to (-1)^{f(x)} |x\rangle$, in order to utilize quantum interference effect. To implement such an n-bit Boolean function, we prepare the n-particle system that has multi-particle interactions among them, from two-particle interactions $I^{(2)}$ up to n-particle interactions $I^{(n)}$ in addition to one-bit rotations, R. A one-time evolution by the Hamiltonian with properly chosen coefficients of the terms in it builds the transformation $U_{\rm q}$ by itself. A Boolean function $f(x)$ used in Deutsch-Jozsa problem[4] has a constraint that $f(x)$ is either constant (at 0 or 1) or balanced. Because of the constraint, the concurrent implementation does not call for the n-particle interaction $I^{(n)}$. In Grover's data search algorithm[6], the function to be implemented $f(x)$ is zero for all x but τ ($f(\tau) = 1$) which we search for. As a general Boolean function, this function requires the system that has all multi-particle interactions in order to be implemented in a concurrent fashion. The transformation $U_{\rm q}$ is constructed by successive applications of elementary gates (one-bit gates and two-bit gates) by the standard sequential implementation with the help of a work bit prepared in the state $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ and another qubit[1]. The number of required gate operations to implement an n-bit Boolean function in the sequential manner scales as $O(n2^n)$ as opposed to a one-time evolution of a system in the case of the concurrent implementation. The function implemented in Grover's algorithm is special in that it only needs one $(n+1)$-bit gate that is constructed by $O(n)$ elementary gates. In Shor's algorithm for prime factoring an odd number N[9], $a^x \bmod N$ (a is a randomly chosen integer relatively prime to N) is encoded in a work bit, as $|x\rangle\,|0\rangle \to |x\rangle\,|a^x \bmod N\rangle$. The construction of a gate array for this transformation requires $O(n^3)$ elementary gates by means of the sequential implementation. Instead of encoding $|a^x \bmod N\rangle$ is the work bit, encoding it in the phase of the control register $|x\rangle$, as $|x\rangle \to \exp\left[i\frac{\pi}{2N}(a^x \bmod N)\right]|x\rangle$, still works[10]. The factor $\frac{\pi}{2N}$ is determined to differentiate all possible values $|a^x\rangle\,N$ takes (between 0 and $N-1$) and also to avoid a destructive interference that ruins the algorithm.

In Simon's algorithm[8], a function to be implemented is $f : \{0,1\}^n \to \{0,1\}^m (m \geq n)$. The sequential implementation of such a function in a work bit is in need of $O(mn2^n)$ elementary gates. As in Shor's algorithm, the function f can be implemented in the phase of the control register $|x\rangle$, as $|x\rangle \to \exp\left[\frac{\pi}{2^{m-1}} f(x)\right]$. To distinguish all 2^m possible values f takes, the phase space of 2π is divided by 2^m (a Boolean function, $f : \{0,1\}^n \to \{0,1\}$, is a special case where $m = 1$. In both Shor's and Simon's algorithms, the concurrent implementation of necessary functions involves all multi-particle interactions $I^{(2)}, \cdots, I^{(n)}$ in the system. A controlled-phase-shift operator $S_{j,k}$ acts on a pair of bits, in this case j-th and k-th qubits of the control register $|x\rangle$. It adds a phase factor $\exp\left(i\frac{\pi}{2^{k-j}}\right)$ only when both x_k and x_j are ones. Only two-particle interactions $I^{(2)}$ between k-th and j-th particles to implement this operator. Such controlled-phase-shift operators on all pairs of qubits ($n(n-1)/2$ paris in total) compose Quantum Fourier transform, which is used in Shor's algorithm, along with the Walsh-Hadamard transformation on each qubit (H_l on l-th qubit)[9]. The Walsh-Hadamard transformations are applied to all qubits, from x_1 to x_n, and controlled-phase-shift operators $S_{l,n}S_{l,n-1}\cdots S_{l,l+1}$ are interleaved between H_l and H_{l+1}. A controlled-phase-shift operator and a product of multiple of them can be concurrently implemented by a one-time evolution of a system that has only two-particle interactions and one-bit rotations. In total, Quantum Fourier transform is constructed by n Walsh-Hadamard transformations and n time-evolutions of a system in a concurrent manner, whereas in the case of the sequential implementation, $n(n-1)/2$ controlled-phase-shift operators are implemented in series.

REFERENCES

1. A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, H. Weinfurter, Phys. Rev. A **52**, 3457 (1995).

2. D. Deutsch, Proc. R. Soc. London A **400**, 97 (1985).

3. R. Cleve, A. Ekert, C. Macciavelo, M. Mosca, Proc. R. Soc. London A **454**, 339 (1998), quant-ph/970816.

4. D. Deutsch, R. Jozsa, Proc. R. Soc. London A **439**, 553 (1992).

5. D. Collins, K. W. Kim, W. C. Holton, Phys. Rev. A **58**, R1633 (1998).

6. L. K. Grover, Phys. Rev. Lett. **79**, 325 (1997).

7. L. K. Grover, Phys. Rev. Lett. **80**, 4329 (1998).

8. D. R. Simon, SIAM J. Comput. **26**, 1474 (1997).

9. P. W. Shor, SIAM J. Comput. **26**, 1484 (1997).

10. Following Shor's original algorithm[9], We then apply the Quantum Fourier transform (12) to the control register $|x\rangle$ in the state (10), and then we obtain
$\frac{1}{q}\sum_{x=0}^{q-1}\sum_{y=0}^{q-1} e^{i\frac{\pi}{2N}(a^x \bmod N)} e^{2\pi i xy/q}\,|y\rangle\,.$

    A measurement projected to $|y\rangle$ has a probability,
$\left|\frac{1}{q}\sum_{x=0}^{q-1} e^{i\frac{\pi}{2N}(a^x \bmod N)} e^{2\pi i xy/q}\right|^2 .$

    Since the order of a is r, $a^x \bmod N$ may be written as a^k, $0 \le k < r$, where $x \equiv k \bmod r$. Therefore, the probability is,
$\left|\sum_{k=0}^{r-1} e^{i\frac{\pi}{2N}(a^k \bmod N)}\frac{1}{q}\sum_{x:a^x\equiv a^k} e^{2\pi i xy/q}\right|^2 = \left|\sum_{k=0}^{r-1} e^{i\frac{\pi}{2N}(a^k \bmod N)}\frac{1}{q}\sum_{b=0}^{\lfloor (q-k-1)/r\rfloor} e^{2\pi i(br+k)yq}\right|^2 .$

    The sum taken over all x satisfying $a^x \equiv a^k \bmod N$, or equivalently $x \equiv k \bmod r$, is replaced by the sum over b, defined by $x = br + k$, and it is decomposed into two terms.

$$\frac{1}{q}\sum_{b=0}^{\lfloor (q-k-1)/r \rfloor} e^{2\pi i(br+k)y/q}$$
$$= \frac{1}{q}\sum_{b=0}^{\lfloor (q-k-1)/r \rfloor} e^{2\pi i b\{ry\}_q/q} + \frac{1}{q}\sum_{b=0}^{\lfloor (q-k-1)/r \rfloor} \left[e^{2\pi i(br+k)cq} - e^{2\pi i b\{ry\}_q/q}\right],$$

where $\{ry\}_q$ is congruent to $ry \bmod q$ ($-q/2 < \{ry\}_q \leq q/2$). The first sum is large only when $\{ry\}_q$ is small. When $|\{ry\}_q| \sim O(r)$, the second sum is $O(1/q)$ and the first term can be approximated as,

$$\frac{1}{r}\int_0^1 \exp\left(2\pi i \frac{\{ry\}_q}{r} u\right) \mathrm{d}u + O\left(\frac{1}{q}\right),$$

where the first term is $O(1/r)$. Neglecting terms $O(1/q)$ ($\ll O(1/r)$), we obtain the probability of observing the state $|y\rangle$,

$$\left|\sum_{k=0}^{r-1} e^{i\frac{\pi}{2N}(a^k \bmod N)}\right|^2 \times \left|\frac{1}{r}\int_0^1 \exp\left(2\pi i \frac{\{ry\}_q}{r} u\right) \mathrm{d}u\right|^2.$$

The second term is the probability of observing the state $|y\rangle$ for a particular value of k. If all possible values of k $(0, 1, \cdots, r-1)$ contribute the probability independently, the total probability is this term multiplied by r. In our case, the first term in the above equation is multiplied instead, due to the k-dependent phase factors. It can be calculated as,

$$\left|\sum_{k=0}^{r-1} e^{i\frac{\pi}{2N}(a^k \bmod N)}\right|^2 = \sum_{k=0}^{r-1}\sum_{k'=0}^{r-1} e^{i\frac{\pi}{2N}(a^k \bmod N)} e^{-i\frac{\pi}{2N}a^{k'}}$$
$$= r + 2\sum_{k=1}^{r-1}\sum_{k'=0}^{k-1} \cos\left[\frac{\pi}{2N}\left((a^k \bmod N) - (a^{k'} \bmod N)\right)\right].$$

Since $-N < (a^k \bmod N) - (a^{k'} \bmod N) < N$, the second term in the above equation is positive. The modified visibility $\left|\sum_{k=0}^{r-1} e^{i\frac{\pi}{2N}(a^k \bmod N)}\right|^2 / r$ is lower-bounded by 1.